Amendment
Appln. No.: 09/274,979

REMARKS

Claims 125-148 and 152-182 now stand in the application, claims 149-152 having been canceled and new claims 180-182 added. Claim 180 is identical to claim 157 but dependent on claim 164, and claims 180 and 181 are new dependent claims. The remaining claims have been amended to more clearly define the invention. Reconsideration of the application and allowance of all claims are respectfully requested in view of the above amendments and the following remarks.

In further review of the claims, applicants noticed that the terminology used in the claims was not always consistent with the terminology used in the specification. The claims have been amended to improve conformity with the disclosure, particularly with respect to the terms "data items" and "type labels." It is also believed that referring to an element as being associated with a type label is more descriptive than simply saying that it "has" a type label. Finally, it is noted that the invention which is the subject of the claims presently standing in this application relates to the use of type labels in association with data items which are either themselves arguments or are contained within data items which are arguments. Noting the definition of "argument" at page 2 of the specification as a data item, and the statement at page 2 of the specification that data items may contain other data items, a "data item of an argument" will be either the argument itself or a data item contained within an argument. This aspect of the claimed invention, i.e., the use of labels in conjunction with data items which are either themselves arguments or are contained within arguments, has now been emphasized in the claims.

Applicants wish to take this opportunity to comment further on the relevance of the NC.Focus paper entitled "The Forthcoming Metadata Revolution" submitted with the response filed March 19, 2003.

There are a number of instances in which the author refers to webMethods' Web Interface Definition Language (WIDL). It is important to note that WIDL is a language for defining a web interface. This is not XML-RPC.

There are also a number of references in the paper to the use of XML in developing distributed object applications. To the extent that this is considered to suggest XML-RPC, (1)

15

Amendment
Appln. No.: 09/274,979

much of what the author describes is attributed to webMethods and applicants reserve the right to submit evidence of derivation from webMethods and/or earlier invention by webMethods, and (2) most importantly, even accepting as prior art whatever the paper does teach about XML-RPC, it does not teach or suggest the invention defined in the claims pending in the present application.

As explained in the present application, conventional RPC requires that the recipient of a message know in advance the data types of the message arguments found in the message. Conventional RPC extracts the data items from a message according to the order in which they appear in the message, and it can do so only because it knows in advance what data types appear at what positions in the message. If one were to use XML to implement conventional RPC, one might have, e.g., a <MESSAGE> element for the message and an <ARGUMENT> element for each input or output argument of the service invoked. Neither MESSAGE nor ARGUMENT provides a type label, as defined in the application and specified in the claims.[1] To interpret this, a recipient must still know in advance what data types occur at what positions, and no benefit is gained. The IIOP 'ANY' data type (see the discussion of this at page 7 of the present application), might appear to be an exception to this rule, because it allows one to attach certain type labels to data, but it is not an exception because an argument of this type does not label itself as the 'ANY' data type -- the recipient must still know this in advance.

The infrastructure for conventional RPC is typically geared toward sequentially serializing message arguments on outbound requests and toward sequentially de-serializing message arguments on inbound replies. So even if one thinks to represent an RPC in XML with type labels and semantic labels, it was not necessarily clear how one would fit this into existing RPC-middleware.

WIDL is a language containing a specification component and an implementation component, neither of which is capable of implementing the present invention. Furthermore,

---

[1] Pages 1-7 of the present application define very clearly a number of terms used throughout the specification and claims, with data types being discussed at page 4.

16

Amendment
Appln. No.: 09/274,979

neither component suggests the present invention because these components conduct RPC messaging in the same way that conventional RPC messaging is conducted.

The first component of WIDL is an interface definition language (an IDL). CORBA IDL and DCE IDL are also interface definition languages. WIDL differs from these other IDLs primarily by being expressed in the standard syntax of XML rather than being expressed in the proprietary syntaxes that CORBA and DCE define. Like all IDLs, the IDL component of WIDL is only capable of expressing the shape of the data that will be sent or received via RPC. This component of WIDL is not capable of expressing the data itself. Moreover, just as with conventional IDLs, when an RPC interface is specified in WIDL, the interface constrains the shape of the data that may be sent or received through the interface, particularly by pre-specifying the data types of the data that one may put in these messages. WIDL makes no statement about whether the message itself associates type labels with its data, but even if the message did associate type labels with its data, a WIDL specification would force the message to contain specific types, thus undermining the value of putting type labels in the message and rendering the type labels useless.

The second component of WIDL is a language for implementing a WIDL-specified interface over the web. This language allows one to transmit messages over HTTP using Common Gateway Interface (CGI), which is part of the HTTP standard and which has no relation to XML. The language also allows one to receive messages via HTTP, where received messages are expressed in either HTML or XML. An RPC component implemented in WIDL is analogous to a stub or a skeleton implemented for CORBA or DCE RPC. All messages sent and all messages received via such a component must conform to the specification for the interface. In the case of WIDL, the arguments in all CGI requests transmitted conform to the specific request data types named by the specification and the arguments in all HTML and XML documents received must conform to the specific response data types named by the specification. Since a WIDL implementation does not recognize and cannot use data type information that a message may contain -- since a WIDL implementation only uses the data type information pre-

17

Amendment
Appln. No.: 09/274,979

specified for the messages in the WIDL specification — a WIDL implementation neither uses nor suggests what is claimed in the present invention.

Thus, even if WIDL were used to develop a distributed object application, the result would still suffer from the drawback that the claimed invention addresses, i.e., the message encoding would still have to know in advance the data types of the argument data.

At the top of page 3,the NC.Focus paper discusses building hierarchical structures of name/value pairs using XML. The author then gives an example of what he means, identifying names such as payer, date, amount, etc., and concludes that "the scope of these name/value pairs only makes sense within the context of an accounting record." But what is unique about the invention claimed herein is that its scope is not limited this way. The present specification defines "semantic" labels, and these labels serve purposes that the NC.Focus paper describes. But the present invention uses what it defines as "type labels" to designate things such as records and arrays. The NC. Focus paper does not suggest the use of the claimed "type labels."

All claims of the present case describe the use of type labels, which is simply not a characteristic suggested in the NC Focus paper, even if combined with what is known from WIDL.

Further examination is respectfully requested.

Respectfully submitted,

David J. Cushing
Registration No. 28,703

SUGHRUE MION, PLLC
2100 Pennsylvania Avenue, N.W.
Washington, D.C. 20037-3213
Telephone: (202) 293-7060
Facsimile: (202) 293-7860
Date: May 7, 2003

18

Amendment
Appln. No.: 09/274,979

APPENDIX
VERSION WITH MARKINGS TO SHOW CHANGES MADE

Please cancel claims 149-152 without prejudice or disclaimer.

Please amend the remaining claims to read as follows:

125.   (Amended) A method of invoking a service at a first machine from a second machine, comprising the steps of generating a service invocation request message at said second machine using a markup language-based message encoding, and transmitting said service invocation request from said second machine, wherein said message includes plural elements representing data items of at least one argument and wherein all of said elements havewhich are associated with element type indicatorslabels selected from an encoding group having a predetermined number of members, with at least two of said members designating elements containing other elements having element associated with type indicatorslabels belonging to said group.

126.   (Amended) A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request generated at a second machine in compliance with a markup language-based message encoding, wherein said message includes plural elements representing data items of at least one argument and wherein all elements in said message have elementassociated with type indicatorslabels selected from an encoding group having a predetermined number of members, with at least two of said members designating elements containing other elements having element-associated with type indicatorslabels belonging to said group; and

invoking said service in response to said request.

127.   (Amended) A method of invoking a service at a first machine from a second machine, comprising the steps of:

19

Amendment
Appln. No.: 09/274,979

generating a service invocation request message at said second machine in
compliance with a markup language-based message encoding, wherein said message
includes plural elements representing data items of at least one argument and ~~wherein all
elements in said message have element~~-associated with type ~~indicators~~labels selected
from an encoding group having a predetermined number of members, including at least a
first ~~element~~-type ~~indicator~~label for designating an element containing lexical data, and a
second ~~element~~-type ~~indicator~~label for designating an element containing ~~a set of~~ other
elements ~~having element~~-associated with type ~~indicators~~labels selected from said group;
and

transmitting said message .

128.   (Amended) A method of invoking a service at a first machine, comprising the
steps of:

receiving at said first machine a service invocation request message generated at a
second machine in compliance with a markup language-based message encoding,
wherein said message includes plural elements representing data items of at least one
argument and ~~wherein all elements in said message have element~~-associated with type
~~indicators~~labels selected from an encoding group having a predetermined number of
members, including at least a first ~~element~~-type ~~indicator~~label for designating an element
containing lexical data, and a second ~~element~~-type ~~indicator~~label for designating an
element containing ~~a set of~~ other elements ~~having element~~-associated with type
~~indicators~~labels selected from said group; and

invoking said service in response to said message.

129.   (Amended) A method of invoking a service at a first machine, said method
comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

20

Amendment
Appln. No.: 09/274,979

transmitting from said first machine a service invocation reply message in compliance with a markup language-based message encoding, wherein said message includes plural elements representing data items of at least one argument and wherein all elements in said message have element associated with type indicatorslabels selected from an encoding group having a predetermined number of members, including at least a first element type indicatorlabel for designating an element containing lexical data, and a second element type indicatorlabel for designating an element containing a set of other elements having element associated with type indicatorslabels selected from said group.

130.   (Amended) A method of invoking a service at a first machine, said method comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in compliance with a markup language-based message encoding, wherein said message includes plural elements representing data items of at least one argument and wherein all elements in said message have element associated with type indicatorslabels selected from an encoding group having a predetermined number of members, including at least a first element type indicatorlabel for designating an element containing lexical data, and a second element type indicatorlabel for designating an element containing a set of other elements having element associated with type indicatorslabels selected from said group.

131.   (Amended) A method according to claim any one of claims 127-130, wherein said encoding group further includes a third element type indicatorlabel for designating an element containing a set of other elements having element associated with type indicatorslabels selected from said group.

132.   (Amended) A method according to claim 131, wherein said encoding group includes a fourth element type indicatorlabel for designating an element containing a set of other elements having element associated with type indicatorslabels selected from said group.

21

Amendment
Appln. No.: 09/274,979

133.   (Amended) A method according to claim 131, wherein said encoding group includes a fourth ~~element~~-type ~~indicator~~label for designating an element uniquely identifying another ~~encoding~~ element within a particular message.

134.   (Amended) A method according to claim 131, wherein said encoding group includes a fourth ~~element~~-type ~~indicator~~label for designating the absence of ~~a~~ data ~~item~~.

135.   (Amended) A method according to claim 133, wherein said encoding group includes a fifth ~~element~~-type ~~indicator~~label for designating the absence of ~~a~~ data ~~item~~.

136.   (Amended) A method according to claim 135, wherein said encoding group includes a sixth ~~element~~-type ~~indicator~~label for designating an element containing ~~a set of~~ other elements ~~having element~~-associated with type ~~indicators~~labels selected from said group.

137.   (Amended) A method according to claim 131, wherein said third ~~element~~-type ~~indicator~~label designates an element containing an n-dimensional array (where n is an integer such that n ≥ 1) of elements ~~having element~~-associated with type ~~indicators~~labels selected from said encoding group.

138.   (Amended) A method according to any one of claims 127-130, wherein said encoding provides a lexical type ~~label~~indicator associated with an element having said first ~~element~~type ~~indicator~~label.

139.   (Amended) A method according to claim 138, wherein an element ~~of~~associated with said first ~~element~~-type ~~indicator~~label with no lexical type ~~label~~indicator is assumed to ~~be a~~contain data of string lexical type~~element~~.

22

Amendment
Appln. No.: 09/274,979

140.   (Amended) A method according to claim 138, wherein said mark-up language is
XML, said ~~element type indicators~~elements are expressed as XML elements, said type labels are
expressed as XML element type names, and said lexical type ~~label~~indicator is expressed as an
XML attribute on ~~said~~an XML element ~~having~~associated with said first ~~element~~ type
~~indicator~~label, with the ~~data~~lexical type of ~~a~~the data ~~item~~contained in said XML element being
designated by the value of said XML attribute.

141.   (Amended) A method according to claim 131, wherein said encoding group
further includes a fourth ~~element~~ type ~~indicator~~label for designating an element representing a
numeric value.

142.   (Amended) A method according to claim 131, wherein said encoding group
includes multiple lexical type ~~names~~labels each designating a respective different type of lexical
data ~~item~~contained in an element ~~having~~associated with said first type ~~indicator~~label.

143.   (Amended) A method according to claim 131, wherein said message further
includes a semantic label for at least one data item~~contained~~of an argument in said message.

144.   (Amended) A method according to claim 143, wherein said mark-up language is
XML, said at least one element is expressed as an XML element, and said semantic label is
~~represented by~~expressed as the value of an XML attribute on ~~the~~said XML element~~ containing
said data item~~.

145.   (Amended) A method of invoking a service at a first machine from a second
machine, said method comprising the steps of:

    generating a service invocation request message at said second machine in
    compliance with a markup language-based message encoding wherein said message
    includes ~~each~~ elements representing data items of at least one argument ~~in said message~~
    ~~is~~and associated with ~~a~~ type ~~indicator~~labels selected from a group of type

23

Amendment
Appln. No.: 09/274,979

~~indicators~~labels, said group including at least an array type ~~indicator~~label indicating that the corresponding element is an array element representing an n-dimensional array containing a plurality of data items, where n is an integer and n ≥ 1, said message including at least one ~~data item which is~~element associated with said array type label and representing a multi-level nested array ~~element~~ where each element nesting level corresponds to a respective dimension of said array ~~element~~; and

transmitting said service invocation request message from said second machine.

146.   (Amended) A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding wherein said message includes ~~each~~ elements representing data items of at least one argument~~in said message is~~ and associated with ~~a~~ type ~~indicator~~labels selected from a group of type ~~indicators~~labels, said group including at least an array type ~~indicator~~label indicating that the corresponding element is an array element representing an n-dimensional array containing a plurality of data items, where n is an integer and n ≥ 1, said message including at least one ~~data item which is~~element associated with said array type label and representing a multi-level nested array ~~element~~ where each element nesting level corresponds to a respective dimension of said array ~~element~~; and

invoking said service in response to said message.

147.   (Amended) A method of invoking a service at a first machine, said method comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in compliance with a markup language-based message encoding wherein said message includes ~~each~~ elements representing data items of at least one argument ~~in said message~~

24

Amendment
Appln. No.: 09/274,979

isand associated with a—type indicatorlabels selected from a group of type indicatorslabels, said group including at least an array type indicatorlabel indicating that the corresponding element is an array element representing an n-dimensional array containing a plurality of data items, where n is an integer and n ≥ 1, said message including at least one data item whichelement associated with said array type label and representing a multi-level nested array element—where each element nesting level corresponds to a respective dimension of said array-element; and

transmitting said service invocation reply message from said second machine.

148.   (Amended) A method of invoking a service at a first machine, said method comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in compliance with a markup language-based message encoding wherein said message includes euch elements representing data items of at least one argument in said message isand associated with a—type indicatorlabels selected from a group of type indicatorslabels, said group including at least an array type indicatorlabel indicating that the corresponding element is an array element representing an n-dimensional array containing a plurality of data items, where n is an integer and n ≥ 1, said message including at least one data item which iselement associated with said array type label and representing a multi-level nested array element—where each element nesting level corresponds to a respective dimension of said array-element.

Claims 149-152 cancelled.

153.   (Amended) A method of invoking a service at a first machine from a second machine, said method comprising the steps of:

generating a service invocation request message at said second machine in compliance with a mark-up language-based message encoding wherein said message

25

Amendment
Appln. No.: 09/274,979

includes ~~each~~ elements representing data items of at least one argument and~~in said~~ ~~message is~~ associated with ~~a~~ type ~~indicator~~ labels selected from a group including at least an array type ~~indicator~~ label indicating that the corresponding element is an array element representing an n-dimensional array containing a plurality of data items, where n is an integer and n ≥ 1, said request message including at least one ~~data item which is~~ element associated with said array type label and representing an array of dimension n and further including ~~a~~ an array label associated with said ~~data item~~ at least one element and ~~designating said data item as having an array type, said encoding~~ requiring that all data items ~~contained~~ represented within said array ~~as direct children~~ have the same type as one another; and

transmitting said service invocation request message from said second machine.

154.　(Amended) A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding wherein said message ~~includes~~ ~~each~~ elements representing data items of at least one argument and~~in said message is~~ associated with ~~a~~ type ~~indicator~~ labels selected from a group including at least an array type ~~indicator~~ label indicating that the corresponding element is an array element representing an n-dimensional array containing a plurality of data items, where n is an integer and n ≥ 1, said request message including at least one ~~data item~~ ~~which is~~ element associated with said array type label and representing an array of dimension n and further including ~~a~~ an array label associated with said ~~data item~~ at least one element and ~~designating said data item as having an array type, said encoding~~ requiring that all data items ~~contained~~ represented within said array ~~as direct children~~ have the same type as one another; and

invoking said service in response to said message.

26

155.   (Amended) A method of invoking a service at a first machine, said method
comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in
compliance with a markup language-based message encoding wherein said message
includeseach elements representing data items of at least one argument andin said
message is associated with a-type indicator-labels selected from a group including at least
an array type indicatorlabel indicating that the corresponding element is an array element
representing an n-dimensional array containing a plurality of data items, where n is an
integer and n ≥ 1, said reply message including at least one data-item-which-iselement
associated with said array type label and representing an array of dimension n and a
further including an array label associated with said data-itemat least one element and
designating said data item as having an array type, said encoding requiring that all data
items represented contained within said array as direct children have the same type as one
another; and

transmitting said service invocation reply message from said second machine.

156.   (Amended) A method of invoking a service at a first machine, said method
comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in
compliance with a markup language-based message encoding wherein said message
includeseach elements representing data items of at least one argument andin said
message is associated with a-type indicator-labels selected from a group including at least
an array type indicatorlabel indicating that the corresponding element is an array element
representing an n-dimensional array containing a plurality of data items, where n is an
integer and n ≥ 1, said reply message including at least one data-item-element which
isassociated with said array type label and representing an array of dimension n and

27

Amendment
Appln. No.: 09/274,979

further including a-an array label associated with said ~~data item~~at least one element and ~~designating said data item as having an array type, said encoding~~requiring that all data items ~~contained~~represented within said array ~~as direct children~~have the same type as one another.

157.   (Amended) A method according to any one of claims 153-156, wherein said array label identifies said same type.

158.   (Amended) A method according to any one of claims ~~149~~153-156, wherein said mark-up language is XML, said at least one element is expressed as an XML element, and said array label is expressed as an XML attribute of said XML element such that the dimension n is given by the value of ~~the~~said XML attribute-.

159.   A method according to any one of claims 145-148 and ~~149~~153-156, wherein said message is an XML document.

160.   (Amended) A method according to any one of claims 145-148, wherein said message includes a type label associated with the nesting element at each said ~~data item~~element nesting level and designating said ~~data item~~nesting element as having an array type.

161.   (Amended) A method according to claim 160, wherein ~~each of~~said ~~second array elements includes~~nesting element contains at least one member element representing a data item, with all data items ~~in each of~~represented by the direct children elements of said ~~second array elements~~nesting element being of the same type as one another.

162.   (Amended) A method according to claim 161, wherein said message includes an array label associated with said nesting element and said array label indicates the type associated with all data items ~~contained in~~represented by said direct children elements~~said array~~.

28

Amendment
Appln. No.: 09/274,979

163.   (Amended) A method according to any one of claims ~~149-156~~145-148, wherein said message includes an array label associated with said nesting element and said array label indicates a value of n but does not indicate a size for each of said n dimensions.

164.   (Amended) A method according to any one of claims 145-148, wherein said message includes ~~a~~ an array label associated with said ~~data item~~at least one element and ~~designating said data item as having an array type, said encoding~~ requiring that all data items contained within said array ~~as direct children~~ have the same type as one another.

165.   (Amended) A method of invoking a service at a first machine from a second machine, said method comprising the steps of:

    generating a service invocation request message at said second machine in compliance with a markup language-based message encoding, wherein said message includes~~each~~ elements representing data items of at least one argument and ~~in said message is~~ associated with ~~an element~~ type ~~indicator~~labels selected from a group including at least first and second ~~element~~ type ~~indicators~~labels, wherein said message associates an element having said first type ~~indicator~~label with an ID value, and wherein said message includes an element ~~having~~associated with said second type ~~indicator~~label which specifies said ID value; and

    transmitting said service invocation request message from said second machine.

166.   (Amended) A method of invoking a service at a first machine, comprising the steps of:

    receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding, wherein said message includes~~each~~ elements representing data items of at least one argument and~~in said message is~~associated with ~~an element~~ type ~~indicator~~labels selected from a group including at least first and second ~~element~~type ~~indicators~~labels , wherein said message associates an element having said first type ~~indicator~~label with an ID value,

29

Amendment
Appln. No.: 09/274,979

and wherein said message includes an element ~~having~~associated with said second type
~~indicator~~label which specifies said ID value; and

invoking said service in response to said message.

167.   (Amended) A method of invoking a service at a first machine, said method
comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in
compliance with a markup language-based message encoding, wherein said message
includes~~each~~ elements representing data items of at least one argument and~~in said~~
~~message is~~ associated with ~~an element~~ type ~~indicator~~labels selected from a group
including at least first and second ~~element~~-type ~~indicators~~labels, wherein said message
associates an element having said first type ~~indicator~~label with an ID value, and wherein
said message includes an element ~~having~~associated with said second type ~~indicator~~label
which specifies said ID value; and

transmitting said service invocation reply message from said second machine.

168.   (Amended) A method of invoking a service at a first machine, said method
comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in
compliance with a markup language-based message encoding, wherein said message
includes~~each~~ elements representing data items of at least one argument and~~in said~~
~~message is~~ associated with ~~an element~~ type ~~indicator~~labels selected from a group
including at least first and second ~~element~~-type ~~indicators~~labels, wherein said message
associates an element ~~having~~associated with said first type ~~indicator~~label with an ID
value, and wherein said message includes an element ~~having~~associated with said second
type ~~indicator~~label which specifies said ID value.

30

Amendment
Appln. No.: 09/274,979

169.   (Amended) A method of invoking a service at a first machine from a second machine, said method comprising the steps of:

generating a service invocation request message at said second machine in compliance with a markup language-based message encoding, wherein said message includes~~each~~ element**s** representing data items of at least one argument and~~in said message is~~ associated with ~~an element~~-type ~~indicator~~labels selected from a group, said group including at least one placeholder ~~element~~-type ~~indicator~~label that designates a placeholder element which represents the absence of data; and

transmitting said service invocation request message from said second machine.

170.   (Amended) A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding, wherein said message includes~~each~~ element**s** representing data items of at least one argument and~~in said message is~~ associated with ~~an element~~-type ~~indicator~~labels selected from a group including at least one placeholder ~~element~~-type ~~indicator~~label that designates a placeholder element which represents the absence of data; and

invoking said service in response to said message.

171.   (Amended) A method of invoking a service at a first machine, said method comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in compliance with a markup language-based message encoding, wherein said message includes~~each~~ element**s** representing data items of at least one argument and~~in said message is~~ associated with ~~an element~~-type ~~indicator~~labels selected from a group, said

31

Amendment
Appln. No.: 09/274,979

group including at least one placeholder ~~element-type indicator~~label that designates a placeholder element which represents the absence of data; and

transmitting said service invocation reply message from said second machine.

172. (Amended) A method of invoking a service at a first machine, said method comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in compliance with a markup language-based message encoding, wherein ~~, wherein~~ said message includes ~~each~~ elements representing data items of at least one argument and ~~in said message is~~ associated with ~~an element~~-type ~~indicator~~labels selected from a group including at least one placeholder ~~element~~-type ~~indicator~~label that designates a placeholder element which represents the absence of data.

173. A method according to any one of claims 169-172, wherein said placeholder element represents a programming language null object reference.

174. (Amended) A method according to any one of claims 169-172, wherein said placeholder element identifies ~~a data item~~an element contained elsewhere in said message.

175. (Amended) A method according to any one of claims ~~165-168~~169-172, wherein said message includes a second type label associated with said placeholder element.

176. (Amended) A method according to any one of claims ~~165-168~~169-172, wherein said message includes a semantic label associated with said placeholder element.

177. A method according to claim 175, wherein said message includes a semantic label associated with said placeholder element.

32

Amendment
Appln. No.: 09/274,979

178.   (Amended) A method according to any one of claims 165-168, wherein said encoding permits any ~~data item~~element in a message to be associated with an ID which uniquely identifies said ~~data item~~element within said message.

179.   (Amended) A method according to claim 178, wherein said mark-up language is XML, said element is expressed as an XML element, and said ID is associated with ~~a data item~~said element via an XML attribute on said ~~data item~~XML element whose value is said ID.

Please add the following new claims:

180.   A method according to claim 164, wherein said array label identifies said same type.

181.   A method according to any one of claims 125-130 or 145-148, wherein all elements in said message designating data items are associated with type labels.

182.   A method according to any one of claims 153-156 or 165-172, wherein all elements of said message representing data items are associated with type labels.

33